

Note : While making/correcting any program make sure that the program always exits gracefully for all conditions.

1.a) Write statements at the lines marked to print the bit-form of a 4 bit integer in human readable order on a little endian machine.

```
void showbits(int a)    [ 4 ]
{
    char *x;
    int i,j;

    for(<add-here>)
    {

        for(j=0;j<8;j++)
        {
            printf("%d",<add-here>);
        }
        printf(" ");
    }
}
```

1.b.) Given the following class definition. Is there anything wrong in the program? If it is correct it and write the correct program. [4]

```
class test
{
    public:
    void f2(int **arr, int sz)
    {
        for(int i=0;i<sz;i++)
            for(int j=0;j<sz;j++)
                cout<<arr[i][j]<<" ";
    }
    void f1()
    {
        int sz;
        cin>>sz;
        int arr[sz][sz];
        f2(arr,sz);
    }
};
```

2.) You have this program typed on your computers also(named 2.c). [6]

What this is trying to do is convert a string of type

"aaabbc" to "aaa3bb2c".

First write down all the possible test cases you would like to give to a program like this.

Then see the code.

There are some serious bugs in it.

Correct those bugs.And submit the problem.

```

void fun(char *a)
{
    int i,count=1,last=a[0],j;

    for(i=0;i<strlen(a);i++)
        if(a[i]!=last) {
            count++;
            last=a[i];
        }
    printf("%d\n",count);
    last=a[strlen(a)-1];
    for(j=strlen(a)+count-1,i=strlen(a)-1;i>=0;last=a[i],j=j-count-1) {
        count=0;

        if( a[i]==a[i-1] ) {
            while( a[i]==last ) {
                count++;
                a[j-count]=a[i];
                i--;
            }
            a[j]=count+48;
        }
        else
            a[j]=a[i--];
    }
}

```

3.a.)After getting placed, CS students of MNNIT had nothing to do. So they decided to play a game. They arranged themselves in a circle, numbered clockwise in increasing order from 0 to A-1, and started playing game according to following rules.

Initially player 0 has a ball. The players take turns passing the ball to each other. At the beginning of the game and before each next pass the following actions are performed. If the player with the ball has already received the ball B times, the game is over. Otherwise, if the player has received the ball q times, he'll pass the ball directly to the person C places to his left if q is even, or C places to his right if q is odd. Given A, B and C, return the number of times that the ball is passed.

Following code simulates the game above. It has a small bug. Find and correct it. [4]

```

int game(int A, int B, int C)
{
    int v[A];
    memset(v,0,sizeof(v));
    int ret=0,ini=0;
    while(1)
    {
        v[ini]++;
        if(v[ini]==B) return ret;
        ret++;
        if(v[ini]%2 != 0)
            ini=(ini+C)%A;
        else
            ini = (ini-C)%A;
    }
}

```

3.b)

Solve the question inside the folder named 3b [6]

4.)

The following program simulates the game of cricket for two players. The array runs contains the runs scored on each ball. Assume that all the runs are scored by running only and no wicket falls (i.e only these two players bat through the innings.)

Batsmen change strike after 6 balls. The following program calculates the runs scored by each batsman 0 and 1 using the array runs that gives the runs scored in each ball.

Initially batsman 0 is on the strike. But somehow the program doesn't give the correct answer. Rectify the error in the code. [8]

```
void finalRuns(int *runs , int n)
{
    int players[2]={0,0};
    int p=0;
    int over=0;
    for(int i=0;i<n;i++)
    {
        if(over==0) p=(p+1)%2;
        if(p==0) players[0]+=runs[i];
        else players[1]+=runs[i];
        if(runs[i]%2!=0)p=(p+1)%2;
    }
    cout<<players[0]<<" "<<players[1]<<endl;
}
```

5.)

The following code intends to find the word 'test' in the string. but something is going wrong . write a test case where it will fail and the method to correct it. [2]

```
#include<stdio.h>
#include<string.h>
int main()
{
    char *s1="This is a test";
    char *s2="test";

    if(strstr(s1,s2)!=NULL)
        printf("'%s' found in '%s'\n",s1,s2);
    else
        printf("'%s' not found in '%s'\n",s1,s2);

    return 0;
}
```

6.)

The following program prints prime numbers less than 100. But some portion of code got missing. complete the portion. [8]

```
void primes(int cap, int t, int composite) {
    int i,j;
```

```

i = t / cap;
j = t % cap;
if(i <= 1)
    primes(cap,t+1,composite);
else if(j == 0)
    primes(cap,t+1,0);
else if(j == i && !composite)
    (printf("%d\t",i), primes(cap,t+1,0));
else if(j > 1 && j < i)
    primes(cap,t+1, composite + !(i % j));
else if(< 1 >)
    <2>
}

```

7.a)

There is a bug in this macro and the way it is used. find and correct it. The other details of the code are irrelevant.

[4]

```
#define ABS(A) ((A)<0)?-(A):(A)
```

```

class compatible
{
public:
int count(int left, int right, int k)
{
    int ret=0;
    for(int i=left;i<=right;i++)
    {
        int sum=0;
        for(int j=1;j<=i;j++)
        {
            if(i%j==0)sum+=j;
        }
        if(ABS(i-sum) <= k ) ret++;
    }
    return ret;
}
};

```

7.b)solve the question inside the folder named 7b. [10]

8.) a) [4]

This code checks if a linked list contains a cycle. Find out and rectify the bugs in it.

```

#include<iostream>
using namespace std;

```

```

struct list
{
    int val;
    list*next;
};

```

```
bool cycle(list*start)
```

```

{
if(start == NULL) return true;
list*forward = start->next;

while(forward!=NULL && forward !=start)
{
forward = forward->next->next; //move two steps in each iteration
start = start->next; //move 1 step in each iteration
}
if(forward == start)return false;
return true;
}

```

8.b.) [2]

Correctly return n bits from index position p from right of an int.

```

unsigned getbits(int x, int p, int n)
{
return (x >> (p-n)) & ~(~0 << n);
}

```

8.c) [4] I have made a atof function which converts strings to double.

Please write down test strings i must check so that i am very sure of my function's success.

9.a.) [4] You know how desperately files 2 & 3 want to use variable a; which they do not have.They have made some mistake by not realizing that in the project there may be other files also using it. Correct the whole project and write the final compilation statement.

file1 :

```

#ifndef Inc
#define Inc
int a;
#endif

```

file2 :#include "file1"
extern int a;

file3 : #include "file2"
extern int a;

gcc -o hello file2.c file3.c

9.b) Do the question from file 9b.c [2]

10.a) [4] What the following code returns.

double doesSomething(double q)

```

{
int count=0;
double a=q/2.0,q1;
do
{
q1=a;
count++;
a=(a+q/a)/2.0;
}while(abs(q1-a)>0.000001);
return a;
}

```

10.b) [6] The following code is a naive implementation of finding common ancestor of two nodes in a Binary Tree. Is it correct? If not please specify and correct the bug.

```
tree* commonParent(tree * S,int p1,int p2)
{
    if(ser(S->right,p1)&& ser(S->right,p2))
        return commonParent(S->right,p1,p2);
    if(ser(S->left,p1)&& ser(S->left,p2))
        return commonParent(S->left,p1,p2);
    return S;
}

bool ser(tree *S,int ptr)
{
    if(S!=NULL)
    {
        if(ptr==S->value)
            return true;
        return(ser(S->left,ptr) || ser(S->right,ptr));
    }
    return false;
}
```

Q11. [2] The following code overloads the << operator as a friend function to the class Array. There is a small bug in the code. Locate and correct the bug.

```
#include<iostream>
using namespace std;

class Array
{
    const static int size=10;
    int arr[size];
public:
    Array()
    {
        for(int i=0;i<size;i++)
            arr[i]=i+1;
    }
    friend ostream operator<<(ostream &os,const Array& ob);
};

ostream operator<<(ostream &os,const Array& ob)
{
    for(int i=0;i<ob.size;i++)
        os<<ob.arr[i]<<" ";
    os<<endl;
}
```

12.) [8]

Consider the following problem

You have a mobile phone with some broken buttons. An array arr represents the state of the buttons of your mobile set. arr[i] = -1 means button i is broken and arr[i]=1 means the button i is intact. Any number that contains the broken digit cannot be dialled using the mobile set.

Given an array arr[10] and a number (int num), function isValid(int num) returns true if the number can be dialled else return false;

The given function isValid(int num) has a serious bug. Find and correct it.

```
#include<iostream>
using namespace std;

#define max 10
int arr[max];

bool isValid(int num)
{
    while(num)
    {
        int a = num%10;
        if(arr[a]==-1)return false;
        num/=10;
    }
    return true;
}
```

14.) [8]

Remove errors from the following code given in the file 14.c

Here's the algorithm. One thing is missing in this algorithm which you need to add to the code apart from correcting it.

Algorithm :

If char copy it.
If a * or / push on stack.
If a + or - remove all * and / from over the stack first , copy them and then push + or - on stack.
If (push on stack
If) remove everything till a (and copy it..

15.) [4] **The following program finds whether in a given array a no. exists more than half the size of the array.**

Give test cases which will fail to give the correct output.

What can be done to get the correct output, given the constraint that you do not have to change the algorithm used.

```
int fun(int a[],int num)
{
    int count,i,last;

    last=a[0];
    count=1;

    for(i=1;i<num;i++)
    {
        if(a[i]==last) count++;
        else {
            if(count==1) {
                last=a[i];
            }
            else count--;
        }
    }
}
```

```

if(count>1) return last;
else return -1;

}

```

16.) [8] This program reverses a linked list k nodes at a time.

i.e 1 2 3 4 5 6 7 8 and k=3 will become 3 2 1 6 5 4 8 7

This program has serious bugs. Correct the program.

(lines marked are potential sources of bugs.)

```

void fun(struct ll **head,int k)
{
    struct ll *tmp,*t1,*t2,*t3,dummy,*tmp2;
    int i;

    dummy.next=*head;

    for(tmp=&dummy;tmp;tmp=tmp2)
    {
        t1=tmp->next; //bug
        t2=t1->next;      //bug
        t3=t3->next;      //bug

        for(i=1;t2 && i<k ; i++)
        {
            t2->next=t1;
            t1=t2,t2=t3;
            t3=t3->next;    //bug
        }

        tmp->next=t2;        //bug
        tmp2=tmp->next;      //bug
        tmp->next=t3;        //bug
    }
}

```

17.a) [4]

Given two pointers in a linked list the given code interchanges the nodes following the pointers.i.e if *p1 & *p2 are pointers p1->next must be p2->next and p2->next must be p1->next.

Correct the code completely.

Write the test case which was not considered earlier.

```

int interchange(List *p1,List *p2)
{
    List *t1,*t2;
    t1=p1->next;
    t2=p2->next;

    p1->next=t2;
    p2->next=t1;

    t1->next=t2->next;
    t2->next=t1->next;
}

```

b.) [2] This is a code to insert nodes in bst. Correct it.


```

int insert(Btree **head,int data)
{
    if(head==NULL) return 0;

    Btree *n,*t;
    n=( Btree *) malloc(sizeof(Btree));
    n->left=n->right=NULL;
    n->data=data;

    if(*head==NULL) {
        *head=n;
        return 1;
    }

    t=*head;
    do {
        if(t->data<data)
            t->right?(t=t->right):t->right=n;
        else t->left?(t=t->left):t->left=n;
    } while( t!=n );

    return 1;
}

```

18.a) [4] The following program does a left to right level order traversal of a bst.

The function prints each level of the tree on a different line. It stops at the line which contains no nodes i.e it does not prints such a line.

Does the following code runs correctly. If not make the corrections.

```

void print_level_order(Btree *head)
{
    int i,j=1,flag=1;
    Queue<Btree *> que;
    que.enqueue(head);

    while(flag) {
        for(i=0;i<j;i++)
        {
            head=que.dequeue();
            printf("%d",head);

            if(head->left || head->right) flag=1;
            que.enqueue(head->left);
            que.enqueue(head->right);
        }

        printf("\n");
        j*=2;
    }
}

```

18.b) [2] What is the following code upto !

```

void fun(Btree *l,void **n)
{
    if(l==NULL) return ;

    fun(l->right,n);
}

```

```

l->right=*n;
*n=l;
fun(l->left,n);
}
//when first called l==head of the tree and *n==NULL

```

19.) [2] The following code is trying but is not able to correctly print the inorder successor of a node in a bst. Please rectify the problem.

```

Btree *inorder_s(Btree *head,Btree *parent,Btree *n)
{
    if(head==NULL) return NULL;

    if(head==n) return s;

    return (inorder_s(head->left,n,head) || inorder_s(head->right,n,parent) );
}

```

20.) [6]

```

void crypt2(char *str,char ch,int *i)
{
    char c;

    if( (*i) < strlen(str) )
    {
        c=str[(*i)++];
        crypt2(str,ch,i);
        str[--(*i)]=c;
    }
    else
        str[--(*i)]=ch;
}

void crypt1(char *str,int *i)
{
    char ch;

    if( *i < strlen(str) )
    {
        ch=str[(*i)++];
        crypt1(str,i);
        crypt2(str,ch,i);
    }
}

```

a) These are two functions crypt1 and crypt2 which are carrying out some particular task. Can you guess what these two functions are doing.

Assume that the function crypt1 is called from the main function as

```

int main()
{
    char *str=<some string>;
    int i=0;
    crypt1(str,i);
}

```

b)if the input to the crypt1 function is "allahabad" what will be the final value of the string after returning back to main.

21) [2] This function takes two integers as input and returns the maximum of two integers.

```
int max(int x,int y)
{
    return x- ( (x-y) >> (sizeof(int)-1) ) * (x-y);
}
```

Is there any problem with the function.If yes,what might be the potential problem?

Q22) [8] Given a 1-D array of size (end-start+1) the following code calculates the max sum that can be achieved by taking the alternative numbers.(We cannot include adjacent numbers in the sum. First and last number are assumed to be adjacent) Report the bug and if possible rectify it. e.g consider the array [1 3 4 0 6 2] the max sum is 1+4+6=11.

```
int maxSum(int *arr,int start,int end)
{
    if(start>end)return 0;
    return max(arr[start]+maxSum(arr,start+2,end-1),maxSum(arr,start+1,end));
}
```

Q23.) [8] You are a piano player and you are going to play in avishkar. you don't like to play all the songs at the same volume, so you decide to change the volume level of your piano before each new song. Before the cultural night begins, you make a list of the number of intervals you will be changing your volume level by before each song. For each volume change, you will decide whether to add that number of intervals to the volume, or subtract it.

You are given a int[]v, the i-th element of which is the number of intervals you will change your volume by before the i-th song. You are also given an int beginLevel, the initial volume of your piano, and an int maxLevel, the highest volume setting of your piano. You cannot change the volume of your piano to a level above maxLevel or below 0 (but exactly 0 is no problem). Return the maximum volume you can use to play the last song. If there is no way to go through the list without exceeding maxLevel or going below 0, return -1.

This is how I implement the function f(..). But my recursion contains a small bug please rectify the bug

```
//All header files included
int v[50]; //assume that the user adjusts fills the value and gives u the size
int v_sz;
int beginLevel,maxLevel;
int f(int ini,int a,int b)
{
    if(a>=v_sz)return -1;
    if(b==0)
    {
        if(ini-v[a] >=0)
            return( f(ini-v[a],a+1,0),f(ini-v[a],a+1,1) );
        return -1;
    }
}
```

```

    }
    else if(b==1)
    {
        if(ini+v[a] <=mn)
            return max( f(ini+v[a],a+1,0), f(ini+v[a],a+1,1) );
        return -1;
    }
}

int main()
{
    //input v;
    //input v_sz //size of v <=50
    //input beginLevel,maxLevel;
    cout<<max(f(bg,0,1),f(bg,0,0)); //Initially I call the function like this;
    return 0;
}

```

Q 24. Do the question given in folder named Q24. [8]